IN THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. **(Currently Amended)** A processor comprising:

a functional unit adapted to execute an instruction issued to it from a dispatch stage; and

a buffer in the dispatch stage coupled to the functional unit adapted to receive from a fetch stage and store a plurality of the instructions before issue to the functional unit and further adapted to store scheduling information associated with the instructions,

wherein the **stored** plurality of the instructions comprise a set of instructions from a loop body, and wherein **the stored scheduling information defines, for each processor cycle in the loop, whether and which of the stored instructions are to be executed by the functional unit in that processor cycle, such that for a first processor execution cycle, the stored scheduling information is used to determine a first one of** the **stored** instructions ~~are~~ **to be selected and** issued to the functional unit from the buffer ~~in accordance with the stored scheduling information so as to cause the functional unit to execute a number of iterations of the loop body,~~ **and such that for a second different processor execution cycle, the stored scheduling information is used to determine a second one of the stored instructions to be selected and issued to the functional unit from the buffer**.

2. (Original) A processor according to claim 1, further comprising:

a decode stage register coupled between the dispatch stage and the functional unit, the functional unit coupled to the decode stage register for executing the instruction issued to the decode stage register from the dispatch stage.

3. (Previously Presented) A processor according to claim 1, further comprising:

control logic coupled to the buffer adapted to cause a certain one of the stored plurality of instructions to be issued to the functional unit in accordance with a loop iteration stage and the stored scheduling information associated with the certain instruction.

4. (Previously Presented) A processor according to claim 3, wherein the control logic is further adapted to cause the certain one of the instructions to be issued in accordance with a cycle within the loop iteration stage and the stored scheduling information associated with the certain instruction.

5. (Previously Presented) A processor according to claim 1, wherein the scheduling information comprises a plurality of loop stage bit masks respectively associated with the plurality of instructions.

6. (Previously Presented) A processor according to claim 3,

wherein the scheduling information comprises a plurality of loop stage bit masks respectively associated with the plurality of instructions, and

wherein the control logic is further adapted to cause the certain one of the instructions to be issued in accordance with the respective one of the loop stage bit masks associated with the certain one of the instructions.

7. (Previously Presented) A processor according to claim 4,

wherein the scheduling information comprises a plurality of loop stage bit masks respectively associated with the plurality of instructions, and

wherein the control logic is further adapted to cause the certain one of the instructions to be issued in accordance with the respective one of the loop stage bit masks associated with the certain one of the instructions.

8. (Previously Presented) A processor according to claim 1, further comprising:

control logic coupled to the buffer, the control logic including:

an iteration initiation interval register for storing a loop iteration initiation parameter;

a loop iteration register for storing a loop iteration parameter; and

a loop cycles register for storing a loop cycles parameter,

wherein the control logic is adapted to cause the plurality of instructions to be issued to the functional unit from the buffer in accordance with the loop iteration initiation parameter, the loop iteration parameter, the loop cycles parameter and the stored scheduling information.

9. (Previously Presented) A processor according to claim 3, wherein the stored set of instructions consists of a kernel set of loop instructions, and wherein the control logic is operative so that the functional unit executes a number of loop iterations of the kernel set of loop instructions, a prologue set of loop instructions different than the stored kernel set of loop instructions, and an epilogue set of loop instructions different than the stored kernel set of loop instructions in accordance with the stored kernel set of loop instructions and received loop parameters.

10. (Previously presented) A processor according to claim 9, wherein the received loop parameters include a loop iteration initiation parameter, a loop iteration parameter and a loop cycles parameter, the control logic including:

an iteration initiation interval register for storing the loop iteration initiation parameter;

a loop iteration register for storing the loop iteration parameter; and

a loop cycles register for storing the loop cycles parameter.

11. (Previously Presented) A processor according to claim 1, wherein the set of instructions consists of a kernel set of loop instructions, the processor further comprising:

control logic coupled to the buffer, the control logic being operative so that the functional unit executes a number of loop iterations of the stored kernel set of loop instructions, a prologue set of loop instructions different than the stored kernel set of loop instructions, and an epilogue set of loop instructions different than the stored kernel set of loop instructions based on the stored kernel set of loop instructions and received loop parameters.

12. (Previously presented) A processor according to claim 11, wherein the received loop parameters include a loop iteration initiation parameter, a loop iteration parameter and a loop cycles parameter, the control logic including:

an iteration initiation interval register for storing the loop iteration initiation parameter;

a loop iteration register for storing the loop iteration parameter; and

a loop cycles register for storing the loop cycles parameter.

13. (Previously Presented) A processor according to claim 2, wherein the set of instructions consists of a kernel set of loop instructions, the processor further comprising:

control logic coupled to the buffer, the control logic being operative so that the functional unit executes a number of loop iterations of the stored kernel set of loop instructions, a prologue set of loop instructions different than the stored kernel set of loop instructions, and an epilogue set of loop instructions different than the stored kernel set of loop instructions based on the stored kernel set of loop instructions and received loop parameters.

14.. (Previously presented) A processor according to claim 13, wherein the received loop parameters include a loop iteration initiation parameter, a loop iteration parameter and a loop cycles parameter, the control logic including:

an iteration initiation interval register for storing the loop iteration initiation parameter;

a loop iteration register for storing the loop iteration parameter; and

a loop cycles register for storing the loop cycles parameter.

15. (Previously presented) A processor according to claim 3, further comprising a fetch unit, wherein the control logic is further operative so that the fetch unit can be shut down during execution a number of iterations of the loop body corresponding to the stored plurality of instructions.

16. (Previously presented) A processor according to claim 8, further comprising a fetch unit, wherein the control logic is further operative so that the fetch unit can be shut down during execution of a number of iterations of the loop body corresponding to the stored plurality of instructions.

17. (Original) A processor according to claim 11, further comprising a fetch unit, wherein the control logic is further operative so that the fetch unit can be shut down during execution of the prologue, kernel and epilogue sets of loop instructions.

18. (Original) A processor according to claim 13, further comprising a fetch unit, wherein the control logic is further operative so that the fetch unit can be shut down during execution of the prologue, kernel and epilogue sets of loop instructions.

19. (Previously presented) A processor according to claim 9, wherein the kernel set of loop instructions comprise modulo variable expansion (MVE) code.

20. (Previously presented) A processor according to claim 11, wherein the kernel set of loop instructions comprise modulo variable expansion(MVE) code.

21. (Previously presented) A processor according to claim 13, wherein the kernel set of loop instructions comprise modulo variable expansion (MVE) code.

22. (Original) A processor according to claim 3, wherein the control logic is further operative to allow interrupts to be handled at the end of a current loop iteration.

23. (Original) A processor according to claim 8, wherein the control logic is further operative to allow interrupts to be handled at the end of a current loop iteration.

24. (Original) A processor according to claim 11, wherein the control logic is further operative to allow interrupts to be handled at the end of a current loop iteration.

25. (Original) A processor according to claim 13, wherein the control logic is operative to allow interrupts to be handled at the end of a current loop iteration.

26. (Previously Presented) A buffer in the dispatch stage of a processor, the buffer being associated with a functional unit that executes instructions issued to it from the dispatch stage, the buffer comprising:

a first portion for receiving from a fetch stage and storing a kernel set of loop instructions; and

a second portion for storing a plurality of modulo schedule stage identifiers respectively associated with the kernel set of loop instructions,

wherein the instructions are issued to the functional unit from the buffer in accordance with the stored modulo schedule stage identifiers so as to cause the functional unit to execute a number of iterations of a loop.

27. (Original) A buffer according to claim 26, wherein the processor further includes control logic, the buffer being coupled to the control logic and the functional unit for causing the kernel set of loop instructions to be issued to the functional unit in accordance with the modulo schedule stage identifiers.

28. (Original) A buffer according to claim 27, wherein the modulo schedule stage identifiers comprise bit fields, the control logic determining whether to issue a certain one of the loop instructions to the functional unit in accordance with a bit position of a set bit in the bit field corresponding to the certain loop instruction.

29. (Original) A buffer according to claim 27, wherein the control logic is operative to cause a number of loop iterations of the kernel set of loop instructions, a prologue set of loop instructions different than the kernel set of loop instructions, and an epilogue set of loop instructions different than the kernel set of loop instructions in accordance with the stored modulo schedule stage identifiers.

30. (Original) A buffer according to claim 26, wherein the loop instructions comprise undecoded instructions, and wherein the processor further includes a decode stage interposed between the functional unit and the buffer for decoding the instructions.

31. (Original) A buffer according to claim 26, wherein the loop instructions comprise decoded instructions in the form of functional unit control signals.

32. **(Currently Amended)** A processor for executing a number of iterations of a loop, the loop including a prologue set of loop instructions, a kernel set of loop instructions and an epilogue set of loop instructions in accordance with a modulo schedule, the processor comprising:

a plurality of functional units; and

a dispatch stage coupled to the functional units for issuing instructions to the functional units, the dispatch stage including:

a plurality of buffers respectively associated with the plurality of functional units, ~~each buffer~~ adapted to store the kernel set of loop instructions, and each buffer further adapted to store a different set of scheduling information associated with the respective functional unit and the stored instructions; and

control logic coupled to the plurality of buffers for causing the stored kernel set of instructions to be selectively issued to the respective functional units in accordance with the stored scheduling information, the control logic being operative so that the functional units execute the number of iterations of the kernel set of loop instructions, the prologue set of loop instructions and the epilogue set of loop instructions based on the stored kernel set of loop instructions and associated sets of scheduling information, wherein during a first processor cycle, the control logic uses the sets of scheduling information to cause a portion of the prologue set of loop instructions to be issued to the respective functional units from the buffers, and wherein during a second processor cycle after the first processor cycle, the control logic uses the sets of scheduling information to cause a portion of the kernel set of loop instructions to be issued to the respective functional units from the buffers, and wherein during a third processor cycle after the second processor cycle, the control logic uses the sets of scheduling information to cause a portion of the epilogue set of loop instructions to be issued to the respective functional units from the buffers.

33. (Original) A processor according to claim 32, further comprising a fetch unit, wherein the control logic is further operative so that the fetch unit can be shut down during execution of the number of iterations of the loop.

34. (Original) A processor according to claim 32, wherein the control logic is further operative to allow interrupts to be handled at the end of a current one of the number of loop iterations, and to complete the number of loop iterations after the interrupt is handled.

35. **(Currently Amended)** A method for executing a number of iterations of a loop in a processor, the loop including a prologue set of loop instructions, a kernel set of loop instructions and an epilogue set of loop instructions, the method comprising:

receiving the kernel set of loop instructions and loop parameters from an instruction stream fetched by the processor;

storing the kernel set of loop instructions at a dispatch stage of the processor;

storing loop parameters ~~in control logic~~ associated with the stored loop instructions in control logic of the processor; and

causing the stored kernel set of instructions to be selectively issued to functional units of the processor in accordance with the stored loop parameters so that the functional units of the processor execute the number of iterations of the kernel set of loop instructions, the prologue set of loop instructions and the epilogue set of loop instructions based on the stored loop instructions.

36. (Previously presented) A method according to claim 35, wherein the step of storing the loop parameters includes:

storing an iteration initiation parameter in an iteration initiation interval register in the control logic;

storing a loop iteration parameter in a loop iteration register in the control logic; and

storing a loop cycles parameter in a loop cycles register in the control logic.

37. (Original) A method according to claim 36, wherein the step of causing the stored kernel set of loop instructions to be selectively issued includes:

adjusting the value in the loop cycles register from the stored loop cycles parameter in accordance with a processor cycle; and

resetting the value in the loop cycles register in accordance with the adjusted value and the stored iteration initiation parameter.

38. (Previously presented) A method according to claim 36, wherein the step of causing the stored kernel set of loop instructions to be selectively issued includes:

adjusting the value in the loop iteration register in accordance with the resetting step; and

completing the issue of loop instructions in accordance with the adjusted value in the loop iteration register.

39. (Original) A method according to claim 35, further comprising:

storing a set of modulo schedule stage identifiers respectively associated with the kernel set of loop instructions, the step of causing the stored kernel set of loop instructions to be selectively issued including the step of comparing the stored loop parameters with the modulo schedule stage identifiers.

40. (Original) A method according to claim 39, wherein the step of comparing the stored loop parameters with the modulo schedule stage identifiers includes the step of indexing to a certain one of the modulo schedule stage identifiers in accordance with a current cycle in the modulo schedule stage indicated by the stored loop parameters, the step of causing the stored kernel set of loop instructions to be selectively issued further including issuing the associated loop instruction to the functional unit if the certain modulo schedule stage identifier indicates that the functional unit is active.

41. (Original) A method according to claim 35, wherein the processor further comprises a fetch unit, the method further comprising:

shutting down the fetch unit during execution of the number of iterations of the loop.

42. (Previously presented) A method according to claim 35, further comprising:

allowing interrupts to be handled at the end of a current one of the number of loop iterations; and

completing the number of loop iterations after the interrupt is handled.

43. **(Currently Amended)** A processor for executing a number of iterations of a loop, the loop including a prologue set of loop instructions, a kernel set of loop instructions and an epilogue set of loop instructions, the processor comprising:

means for receiving the kernel set of loop instructions and loop parameters from an instruction stream fetched by the processor;

means for storing the kernel set of loop instructions at a dispatch stage of the processor;

means for storing loop parameters ~~in control logic~~ associated with the stored loop instructions in control logic of the processor; and

means for causing the stored kernel set of instructions to be selectively issued to functional units of the processor in accordance with the stored loop parameters so that the functional units of the processor execute the number of iterations of the kernel set of loop instructions, the prologue set of loop instructions and the epilogue set of loop instructions based on the stored loop instructions.

44. (Previously presented) A processor according to claim 43, wherein the means for storing the loop parameters includes:

means for storing an iteration initiation parameter in an iteration initiation interval register in the control logic;

means for storing a loop iteration parameter in a loop iteration register in the control logic; and

means for storing a loop cycles parameter in a loop cycles register in the control logic.

45. (Original) A processor according to claim 44, wherein the means for causing the stored kernel set of loop instructions to be selectively issued includes:

means for adjusting the value in the loop cycles register from the stored loop cycles parameter in accordance with a processor cycle; and

means for resetting the value in the loop cycles register in accordance with the adjusted value and the stored iteration initiation parameter.

46. (Original) A processor according to claim 45, wherein the means for causing the stored kernel set of loop instructions to be selectively issued includes:

means for adjusting the value in the loop iteration register in accordance with the operation of the resetting means; and

means for completing the issue of loop instructions in accordance with the adjusted value in the loop iteration register.

47. (Original) A processor according to claim 43, further comprising:

means for storing a set of modulo schedule stage identifiers respectively associated with the kernel set of loop instructions, the means for causing the stored kernel set of loop instructions to be selectively issued including means for comparing the stored loop parameters with the modulo schedule stage identifiers.

48. (Original) A processor according to claim 47, wherein the means for comparing the stored loop parameters with the modulo schedule stage identifiers includes means for indexing to a certain one of the modulo schedule stage identifiers in accordance with a current cycle in the modulo schedule stage indicated by the stored loop parameters, the means for causing the stored kernel set of loop instructions to be selectively issued further including means for issuing the associated loop instruction to the functional unit if the certain modulo schedule stage identifier indicates that the functional unit is active.

49. (Original) A processor according to claim 43, wherein the processor further comprises a fetch unit, the processor further comprising:

means for shutting down the fetch unit during execution of the number of iterations of the loop.

50. (Previously presented) A processor according to claim 43, further comprising:

means for allowing interrupts to be handled at the end of a current one of the number of loop iterations; and

means for completing the number of loop iterations after the interrupt is handled.

51. (Canceled)

52. (Canceled)

53. (Previously Presented) A processor comprising:

a functional unit;

a buffer that is coupled to provide instructions for execution by the functional unit;

control logic coupled to the buffer, the control logic being adapted to:

receive loop parameters associated with loop instructions issued to the functional unit from a fetch stage;

cause a kernel set of the loop instructions issued to the functional unit to be stored in the buffer in accordance with the received loop parameters;

cause the functional unit to execute a number of iterations of the kernel set of the loop instructions based on the received loop parameters;

cause the functional unit to further execute a prologue set of the loop instructions different from the stored kernel set of instructions based on the stored kernel set of instructions and the received loop parameters; and

cause the functional unit to further execute an epilogue set of the loop instructions different from the stored kernel set of instructions based on the stored kernel set of instructions and the received loop parameters.

54. (Previously Presented) A processor according to claim 1, wherein the buffer is further adapted to subsequently receive from the fetch stage and store a second different plurality of the instructions, thereby overwriting the previously stored plurality of instructions, before issue to the functional unit and further adapted to store second different scheduling information associated with the second plurality of instructions, wherein the second plurality of the

instructions comprise a set of instructions from a second different loop body, and wherein the second instructions are issued to the functional unit from the buffer in accordance with the stored second scheduling information so as to cause the functional unit to execute a second number of iterations of the second loop body.